Noname manuscript No.
(will be inserted by the editor)

A Survey of Domain Adaptation for Statistical Machine Translation

Hoang Cuong · Khalil Sima'an

Received: date / Accepted: date

Abstract Differences in domains of language use between training data and test data have often been reported to result in performance degradation for phrase-based machine translation models. Throughout the past decade or so, a large body of work aimed at exploring domain adaptation methods to improve system performance in the face of such domain differences. This paper provides a systematic survey of domain adaptation methods for phrase-based machine translation systems. The survey starts out with outlining the sources of errors in various components of phrase-based models due to domain change, including lexical selection, reordering and optimization. Subsequently, it outlines the different research lines to domain adaptation in the literature, and surveys the existing work within these research lines, discussing how these approaches differ and how they relate to each other.

1 Introduction

Machine Translation systems are often applied in settings where the test data might be sampled from a distribution that differs from the training data, usualy due to a different domains of language use. This domain mismatch between train and test data often leads to performance degradation, often due to lexical differences between the domains. When a word in the test data is found in the training data, its most suitable translation in the test domain could be different from that in the training domain. For example, when translating from English to Russian, the most natural translation for the word '*code*' would be 'шифр', 'закон' or 'программа' if we consider *cryptography*, *legal* and *software development* domains, respectively. Given the parallel training data originating from one of those domains, training an MT system on the data would produce a rather suboptimal translation for the other domains.

ILLC, University of Amsterdam
E-mail: {c.hoang,k.simaan}@uva.nl

Surprisingly, degradation of translation quality is observed even when we train an MT system on large heterogeneous corpora (e.g. EuroParl, Common Crawl Corpus, UN Corpus, News Commentary) (Shah et al, 2012; Carpuat et al, 2014; Cuong et al, 2016). For instance, Axelrod et al (2011) show that when it comes to a domain-specific task, a small percentage but well-selected data can outperform the full heterogeneous dataset for training MT systems. Shah et al (2010) show that it would benefit from training word alignment with weighting sentence pairs according to their relevance to a domain-specific task.

In this paper, we provide a comprehensive survey of domain adaptation for statistical machine translation (SMT), aimed particularly at phrase-based systems Koehn (2010). The survey is organized as follows. We first introduce SMT in general, with a focus on aspects of SMT relevant to domain adaptation (Sections 2 and 3).[1] The survey identifies components that need to be adapted when an SMT system is applied to new domains (Section 4). We explain what may go wrong with translation: we analyze potential sources of translation errors and provide an explanation why each specific type of error may happen.

Subsequently, we present a general picture of domain adaptation for SMT where we outline the main general approaches (Section 5). A major part focuses on the induction (Section 6) and combination (Section 7) of domain-focused phrase translation tables, lexical weights and reordering probabilities. The induction of domain-focused sparse features and word alignment probabilities are discussed in Section 8.1 and Section 8.2.

Finally we also cover several practical adaptation scenarios. One of the scenarios is adapting an existing system to multiple specific domains at the same time (Section 8.3). Another scenario is embedding an SMT system into a Cross Lingual Information Retrieval system (i.e. automatically translating queries into different languages, so that a search engine can return search results in the corresponding languages) (Section 8.4). We also discuss how web-based translation services such as Bing Translator and Google Translate can be improved when the domain of a new request is not a priori known. Specifically, we cover cache-based adaptive models (Section 8.5) and rewarding domain invariance for adaptation (Section 8.6).

2 Statistical machine translation

In statistical machine translation (SMT) we aim to translate a source (foreign) sentence $\mathbf{f}$ into a sentence in the target language $\mathbf{e}$. Among the target translation hypotheses, the translation hypothesis $\hat{\mathbf{e}}$ with the highest probability given the source sentence is selected as follows:

$$\hat{\mathbf{e}} = \text{argmax}_{\mathbf{e}}\{P(\mathbf{e}|\ \mathbf{f})\} = \text{argmax}_{\mathbf{e}}\{P(\mathbf{e})P(\mathbf{f}|\ \mathbf{e})\}. \quad (1)$$

This approach to modeling translation is referred to as the noisy-channel framework. The architecture of the framework includes two components: the

[1] Readers may refer to (Koehn, 2010) or (Lopez, 2008) for a comprehensive survey of SMT in general.

translation model (i.e. $P(\mathbf{f}|\ \mathbf{e})$) and the language model (i.e. $P(\mathbf{e})$). The approach was first proposed by Brown et al (1993).

A more powerful approach exploits a log-linear formulation, More formally, where the posterior probability $P(\mathbf{e}|\ \mathbf{f})$ is modeled with a set of M feature functions $\boldsymbol{\phi}(\mathbf{e},\ \mathbf{f}) = \{\phi_1(\mathbf{e},\ \mathbf{f}),\ \ldots,\ \phi_M(\mathbf{e},\ \mathbf{f})\}$ with model parameters $\mathbf{w} = \{w_1,\ \ldots,\ w_M\}$ as follows:

$$P(\mathbf{e}|\ \mathbf{f}) \propto \exp(\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{e},\ \mathbf{f})). \tag{2}$$

Under this framework, we obtain the following decision rule:

$$\hat{\mathbf{e}} = \text{argmax}_{\mathbf{e}}\, \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{e},\ \mathbf{f}). \tag{3}$$

The decision rule is simple as we can safely ignore the daunting normalization factor.

The model was first proposed by Och and Ney (2002), forming the basis of phrase-based SMT systems. It is straightforward to see that this framework contains the noisy-channel framework as a special case (Och and Ney, 2002). Its advantage lies in its flexibility, relative to the noisy-channel framework. One can extend a basic SMT system with translation and language models by including arbitrary feature functions of the source and the target sentences. There are many possibilities for defining feature functions that help the SMT system to improve translation, such as linguistic features, word and phrase penalties, reordering features, rule counting. Simply adding feature functions from the target to source language also often improves translation.

Learning the model parameters $\mathbf{w} = \{w_1,\ \ldots,\ w_M\}$) using a held-out development set is crucial to improving translation. In principle, training for log-linear models can be done using maximum likelihood or related criteria (e.g. *cross-entropy*, *perplexity*). Such an objective function is convex, and global optimization is possible. The main difficulty, however, is that we need to compute the normalization factor during learning. This is intractable, as we cannot explore the full space of all translation hypotheses for each translation input. In practice, the normalization factor is computed using an N-best list of top N translation hypotheses or a lattice (Macherey et al, 2008).[2]

Optimizing an SMT system using maximum likelihood or related criteria has a loose relation to the translation quality on unseen text (Och, 2003). There is a need to directly incorporate translation accuracy on a held-out development set into the optimization. The optimization of the system parameters in this way is now a fundamental part of modern SMT systems. Numerous optimization methods were proposed in the literature, such as MERT (Och, 2003), MIRA (Watanabe et al, 2007; Chiang et al, 2008; Cherry and Foster, 2012), Pairwise Ranked Optimization (Hopkins and May, 2011). Readers may refer to Neubig and Watanabe (2016) for a comprehensive survey of system optimization methods in general.

The latter SMT framework has two notable shortcomings that make the problem of domain adaptation for SMT even more challenging:

[2] As a side note, the size of N-best does not seem to have a significant impact on adaptation (e.g. see Bertoldi and Federico (2009)).

– First, having more translation features significantly increases the difficulty of the optimization. Specifically, having more feature dimensions requires a much larger held-out development set for system optimization, as shown in Waite and Byrne (2015). This is an issue in domain adaptation for SMT because creating such an in-domain held-out development dataset is expensive.
– Second, log-linear models try to separate good and bad translation hypotheses using a linear hyper-plane. This is potentially problematic, as interactions between domain-specific features can be complex. It may be necessary to perform preprocessing steps over the feature space to produce a feature set that is less prone to non-linearities (Clark et al, 2014; Liu et al, 2013). However, methods tailored to such a special treatment are quite sophisticated and not widely deployed in practice.

3 Phrase-based SMT system

There are many types of translation systems that have been built in the past, for example:

– Syntax-based translation (Yamada and Knight, 2001),
– Phrase-based SMT system (Koehn et al, 2003; Och and Ney, 2004),
– Hierarchical phrase-based SMT system (Chiang, 2005, 2007),
– Syntactic phrase-based SMT system (Quirk et al, 2005; Quirk and Menezes, 2006).

This paper focuses on phrase-based SMT systems (Koehn et al, 2003; Och and Ney, 2004).

3.1 Model

A standard phrase-based SMT system has various dense feature functions (i.e. highly informative feature functions) estimated at phrase level. Three of the most important translation models are a phrase-based model $\phi_{TM}(\mathbf{e}, \mathbf{f})$, lexical weight $\phi_{LW}(\mathbf{e}, \mathbf{f})$, and reordering model $\phi_{RM}(\mathbf{e}, \mathbf{f})$. A common domain adaptation strategy for SMT is directly adapting these models. We thus describe them in detail below.

– **Phrase-based model**: At the core of a phrase-based SMT system is the phrase-based model, which aims at modeling translation of sentence pairs at phrase level. Given an input sentence $\mathbf{f}$, let us assume a sequence of target-language phrases $\mathbf{e} = (\tilde{e}_1, \tilde{e}_2, \cdots, \tilde{e}_n)$ is currently hypothesized by the decoder. Let us also assume we are provided with a phrase alignment $\mathbf{a} = (a_1, a_2, \cdots, a_n)$ that defines a source $\tilde{f}_{a_i}$ for each translated phrase $\tilde{e}_i$.

The model is estimated as follows

$$\phi_{TM}(\mathbf{e},\mathbf{f}) = \log P_{TM}(\mathbf{e}|\ \mathbf{f}) = \log \prod_{i=1}^{n} P(\tilde{e}_i|\ \tilde{f}_{a_i}) = \sum_{i=1}^{n} \log P(\tilde{e}_i|\ \tilde{f}_{a_i}) \quad (4)$$

– **Lexical weight**: The lexical weight provide smoother estimates for probabilities of phrase pairs. The model is estimated as follows:

$$\phi_{LW}(\mathbf{e},\mathbf{f}) = \log P_{LW}(\mathbf{e}|\ \mathbf{f}) = \log \prod_{i=1}^{n} P(\tilde{e}_i|\ \tilde{f}_{a_i},\ a_i) = \sum_{i=1}^{n} \log P(\tilde{e}_i|\ \tilde{f}_{a_i},\ a_i) \quad (5)$$

Here, the distribution $P(\tilde{e}_i|\ \tilde{f}_{a_i},\ a_i)$ is computed based on lexical probabilities, $P(e \mid f)$ between words $\langle e,\ f\rangle$ in a phrase pair $\langle \tilde{e}_i,\ \tilde{f}_{a_i}\rangle$. Different models have a slightly different way of computing $P(\tilde{e}_i|\ \tilde{f}_{a_i},\ a_i)$. A typical estimate of $P(\tilde{e}_i|\ \tilde{f}_{a_i},\ a_i)$ (Koehn et al (2003)) is as follows:

$$P(\tilde{e}_i|\ \tilde{f}_{a_i},\ a_i) = \prod_{i=1}^{|\tilde{e}_i|} \frac{1}{|\{j|(j,\ k) \in a_i\}|} \sum_{(j,\ k)\in a_i} P(\tilde{e}_i^k|\ \tilde{f}_{a_i}^j). \quad (6)$$

Here,
- $\tilde{e}_i^k$: word at position k in target phrase $\tilde{e}_i$,
- $\tilde{f}_{a_i}^j$: word at position j in source phrase $\tilde{f}_{a_i}$.
- $|\tilde{e}_i|$: length of phrase $\tilde{e}_i$
- $|\{j|(j,\ k) \in a_i\}|$: the number of source words that each target word at position k in phrase $\tilde{e}_i$ aligns to.

– **Reordering model**: Such phrase-based models and lexical weight are not meant for handling word/phrase order phenomena between languages. For state-of-the-art phrase-based SMT systems, integrating *lexicalized reordering models* (Tillmann, 2004; Koehn et al, 2007; Galley and Manning, 2008) can be considered a must. These models estimate the probability of a sequence of orientations $\mathbf{O} = (o_1,\ o_2,\ \ldots,\ o_n)$ as follows:

$$\phi_{RM}(\mathbf{e},\mathbf{f},\mathbf{O}) = \log P_{RM}(\mathbf{O}|\ \mathbf{e},\ \mathbf{f}) = \log \prod_{i=1}^{n} P(o_i|\tilde{e}_i,\ \tilde{f}_{a_i},\ a_{i-1},\ a_{i-2}) = \sum_{i=1}^{n} \log P(o_i|\ \tilde{e}_i,\ \tilde{f}_{a_i},\ a_{i-1},\ a_{i-2}) \quad (7)$$

Here, each orientation o_i takes possible values $\{M, S, D\}$, representing how likely a phrase is to directly follow a previous phrase (***M****onotone*), to swap positions with it (***S****wap*), or to be not adjacent to it (***D****iscontinous*).

Beside these three types of dense translation features, there are also penalties for word, phrase and distance-based reordering. Those are the basic translation features that form a phrase-based SMT system (beside the language model).

A phrase-based SMT system can be also augmented with millions of sparse feature functions (e.g. phrase features (Chiang et al, 2009; Simianer et al,

2012), lexical features (Watanabe et al, 2007; Chiang et al, 2009), syntax-based features (Blunsom and Osborne, 2008; Marton and Resnik, 2008)). It is possible to induce sparse features using a large portion of the parallel training data. However, scaling training to large data requires extensive additional efforts (e.g. see Yu et al (2013)). Models employing sparse features are often trained using a small held-out development set in practice.

3.2 Training

The most common approach to training a phrase-based SMT system is using relative frequency estimation. We take phrase translation scores as an example. To compute $P(\tilde{e}|\ \tilde{f})$, we first count the number of times phrase $\tilde{e}$ aligns to phrase $\tilde{f}$ in the parallel training data, before normalizing into probability by dividing by the total number of possible alignments to $\tilde{f}$:

$$P(\tilde{e}|\ \tilde{f}) = \frac{c(\tilde{e},\ \tilde{f})}{\sum_{\tilde{e}'} c(\tilde{e}',\ \tilde{f})} \tag{8}$$

This distribution, however, does not necessarily maximize the likelihood of the parallel training data. This is similar to Data Oriented Parsing (DOP) method (Bod et al, 2003) in parsing, which hypothesizes a distribution over many possible derivations of each training example from subtrees of varying sizes.

The key to the training is extracting bilingual phrases from bilingual data. The standard way is to rely on the word-aligned training data, using one of heuristic methods such as *grow-diag-final-and*, *grow-diag-final* or *final* (Koehn et al, 2003)).

Word alignment

We now discuss how to create word-aligned training data. Given a parallel sentence, we look for the most probable alignment between words, $\hat{\mathbf{a}}$, as follows:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e}). \tag{9}$$

The idea of word alignment can be traced back to Brown et al (1990). The degree of difficulty of the search in Eq. 9 depends on the underlying independence assumptions. Even now, twenty years since IBM Models (Brown et al, 1993) and the HMM-based alignment model (Vogel et al, 1996), word alignment is still an active research topic (Simion et al, 2013; Tamura et al, 2014; Chang et al, 2014; Shen et al, 2015; Wang et al, 2015; Liu et al, 2015).

We now briefly review the HMM alignment model (Vogel et al, 1996), which is one of the most popular and widely used alignment models. The generative story of the model is shown in Figure 1. The latent states rely on the target language words and generate source language words.

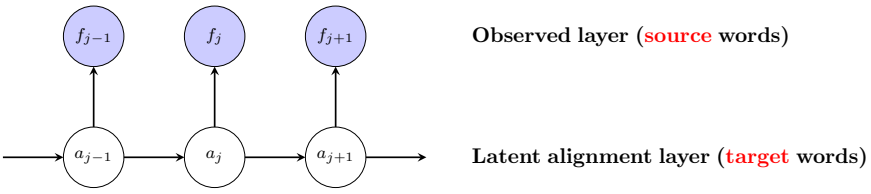

Fig. 1 HMM alignment model with observed and latent alignment layers.

E-step

$$c(f|\ e;\ \mathbf{f},\ \mathbf{e}) = \sum_{\mathbf{a}} \frac{P^{(c)}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})}{P^{(c)}(\mathbf{f}|\ \mathbf{e})} \sum_{j=1}^{J} \delta(f,\ f_j) \sum_{i=0}^{I} \delta(e,\ e_i) \tag{11}$$

$$c(i|\ i';\ \mathbf{f},\ \mathbf{e}) = \sum_{\mathbf{a}} \frac{P^{(c)}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})}{P^{(c)}(\mathbf{f}|\ \mathbf{e})} \sum_{j=1}^{J} \delta(a_j,\ i)\delta(a_{j-1},\ i') \tag{12}$$

M-step

$$P^{(+)}(f|e) = \frac{\sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(f|e;\mathbf{f},\ \mathbf{e})}{\sum_f \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(f|\ e;\ \mathbf{f},\ \mathbf{e})},\ P^{(+)}(i|i') = \frac{\sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(i|i';\mathbf{f},\ \mathbf{e})}{\sum_i \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(i|\ i';\ \mathbf{f},\ \mathbf{e})} \tag{13}$$

Fig. 2 Pseudocode for the training algorithm for the HMM alignment model. Note that $P^{(c)}$ denotes current iteration estimates, $P^{(+)}$ denotes the re-estimates and δ denotes the Kronecker delta function. Note that $P(\cdot|\ \cdot) = \sum_{\mathbf{a}} P(\cdot,\ \mathbf{a}|\ \cdot)$ and it can be computed efficiently using dynamic programming.

Formally, let us assume the target sentence $\mathbf{e}$ contains I words as $\mathbf{e} = (e_1,\ \ldots,\ e_I)$ and the source sentence $\mathbf{f}$ contains J words as $\mathbf{f} = (f_1,\ \ldots,\ f_J)$. For an alignment $\mathbf{a} = (a_1,\ \ldots,\ a_J)$ of the sentence pair $\langle \mathbf{e},\ \mathbf{f}\rangle$, the model factors $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})$ into the word translation and transition probabilities as follows:

$$P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e}) = \prod_{j=1}^{J} P(f_j|\ e_{a_j}) P(a_j|\ a_{j-1}). \tag{10}$$

Here, $P(f_j|\ e_{a_j})$ represents word translation probabilities and $P(a_j|\ a_{j-1})$ represents word transition probabilities. In practice $P(a_j|\ a_{j-1})$ depends only on the distance $(a_j - a_{j-1})$. Note also that the first-order dependency model is an extension of the uniform dependency model of IBM Model 1 and zero-order dependency model of IBM model 2. With the HMM alignment model, the most probable alignment, $\hat{\mathbf{a}}$ for each sentence pair can be computed efficiently using the Viterbi algorithm.

The HMM alignment model has two kinds of parameters - word translation and transition probabilities. Designing the Expectation Maximization (EM) algorithm (Dempster et al, 1977) for training the model is straightforward (Vogel et al, 1996). For the sake of completeness we present the algorithm in detail. We use $c(f|\ e;\ \mathbf{f},\ \mathbf{e})$ to denote the expected count of word e aligns to word f. We also use $c(i|\ i';\ \mathbf{f},\ \mathbf{e})$ to denote the expected counts of two certain consecutive source words j and $j-1$ align to two target words i and i' respectively. Figure 2 presents the algorithm.

Does word alignment suffer from domain mismatch? A domain mismatch could have a negative impact on word alignment accuracy, for example:

- Word alignment models, like any statistical models, suffer from lack of in-domain data for training (Shah et al, 2010; Duh et al, 2010; Gao et al, 2011).
- The insensitivity of existing word alignment models to domains often yields suboptimal results on large heterogeneous data (Gao et al, 2011; Cuong and Sima'an, 2015).

In Section 8.2 we discuss this aspect in detail.

3.3 Decoding

Decoding for phrase-based SMT system is a difficult problem. The search can be done by various approaches (e.g. beam search (Koehn, 2004), exact decoding (Chang and Collins, 2011; Aziz et al, 2014)). Among the approaches, beam search is probably the most popular decoding framework for phrase-based SMT systems. Starting from an initial hypothesis, given an input string of words, a number of phrase translations could be applied to expand the current hypothesis. The expansion is continued until all words are marked as translated.

Beam search heuristically prunes the search space, and, as a result, the search is inexact and search errors can occur as the best scoring hypothesis is not necessarily optimal in terms of given model parameters. Extensive prior work on minimum Bayes risk (MBR) objectives (e.g. see (Kumar and Byrne, 2004)) can potentially mitigate the issue. MBR methods select translations that are less "risky" by taking the uncertainty in model predictions into account. Section 8.6 discusses a link between MBR and domain adaptation for SMT.

4 Translation errors when applied to new domains

Applying a phrase-based SMT system to new domains produces suboptimal translation in practice, e.g. Newswire (Foster et al, 2013), Medical (Irvine et al, 2013b), Patents (Wäschle and Riezler, 2012), Transcribed Lectures (Federico et al, 2012), Web Blog (Su et al, 2012; Foster et al, 2013), TED Talks (Duh et al, 2010; Mansour et al, 2011; Hasler et al, 2014), Subtitles (Irvine et al, 2013b), or Web Queries (Nikoulina et al, 2012). This section reviews different sources of translation errors when applied to new domains.

4.1 Lexical selection

Lexical selection appears to be the most common source of errors (Irvine et al, 2013a; Wees et al, 2015). We present some examples in Table 1. Here, we train

	English-Spanish (Task: Consumer and Industrial Electronics)
Input	*El reproductor puede reproducir señales de audio grabadas en mix-mode cd, cd-g, cd-extra y cd text.*
Human Translation	*The player **can play back audio signals** recorded in mix-mode cd, cd-g, cd-extra and cd text.*
SMT Output	*The player **can reproduce signs of audio** recorded in mix-mode cd, cd-g, cd-extra and cd text.*
Input	*Se puede crear un archivo autodescodificable cuando el archivo codificado se abre con la contraseña maestra.*
Human Translation	*A self-decrypting file can be created when the encrypted file is **opened with** the **master password**.*
SMT Output	*To create an file autodescodificable when the file codified **commenced with** the **password teacher**.*
Input	*Repite todas las pistas (únicamente cds de vídeo sin pbc)*
Human Translation	*Repeat **all tracks** (non-pbc video cds only)*
SMT Output	*Repeated **all avenues** (only cds video without pbc)*

Table 1 Translation errors on unseen domain.

a standard phrase-based SMT system for English-Spanish on a large dataset combined from multiple resources including EuroParl, Common Crawl Corpus, UN Corpus, News Commentary. We then apply the system to a new domain of "*Consumer and Industrial Electronics*". As shown in Table 1, incorrect translations are "*can reproduce signs of audio*" instead of "*can play back audio signals*", "*password teacher*" instead of "*master password*", "*commenced with*" instead of "*opened with*" file, "Repeated *all avenues*" instead of "Repeat *all tracks*".

An important question is what went wrong with lexical selection, or, in other words, what made a phrase-based SMT system suffer from degradation of lexical translation quality on new domains. Two main different error types that cause the degradation are as follows (Irvine et al, 2013a):[3]

- SEEN/SENSE: an incorrect translation for unobserved source language words and an incorrect translation because of known source-language words but with unobserved target words in the parallel training data.
- SCORE: an incorrect translation for which the system goes for an incorrect translation path (i.e. incorrect ranking).

The majority of degradation of lexical translation quality is due to SEEN and SENSE errors. However, it is important to understand that improving coverage does not necessarily result in improved translation quality. This leads to the error type of SCORE, which is perhaps a much harder problem to address.

To provide a better understanding of the SCORE error, let us step back and reconsider how SMT models are estimated (Figure 3). Statistical translation models are trained without integrating (likely hidden) domain information of the bilingual data. This results in coarse and domain-confused translation statistics that reflect translation preferences *aggregated* over different translation options with respect to different domains. Some translation options are

[3] In principle, search errors caused by a decoding algorithm can be a factor. The contribution of the factor to degradation of lexical translation quality, however, is minor, as shown in Irvine et al (2013a).

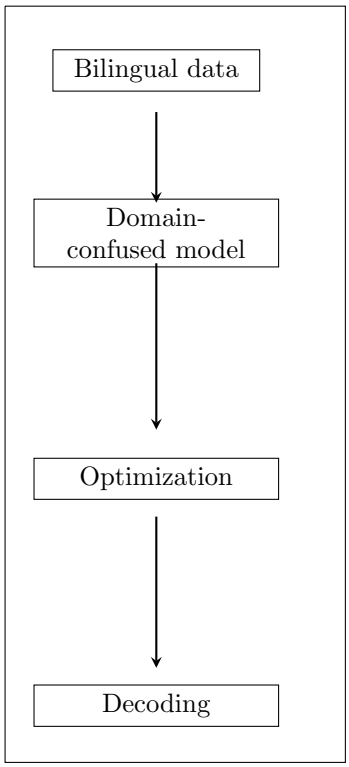

Fig. 3 Statistical translation framework.

more popular than others for a specific word or phrase in general. When it comes to a specific domain, however, it is likely that one of the rare translation options would be the most relevant one. A standard phrase-based SMT system is unlikely to be able to provide such a translation in this case, given that resulting domain-confused statistics are not expressive enough as they do not take domain information into account.

4.2 Reordering

Different from the lexical selection, it is not clear that reordering model adaptation improves translation. There is some evidence supporting this hypothesis, notably from (Chen et al, 2013a) and (Zhang et al, 2015). Chen et al (2013a) show that there are two potential reasons for an improvement in translation quality caused by reordering model adaptation:

- Some corpora may be better for training reordering models than others.
- There exists domain-dependent differences in reordering.

The first statement is intuitively plausible. Some data may contain noisy parallel sentences (e.g. comparable data), or simply too short sentence pairs

English-German (Task: Legal)	
Tuning Scenario	**BLEU↑**
In-domain (**Legal**)	28.8
Mixed-domains (**Including Legal**)	28.5
Mixed-domains (**Exclude Legal**)	28.3

Table 2 Degradation of translation quality on a domain-specific translation task with different tuning scenarios.

(e.g. Subtitles, Search Queries). This has a negative impact on parameter estimates (i.e. less accurate estimates).

Meanwhile, it is not at all obvious that reordering of phrase pairs is particularly domain-specific. Chen et al (2013a) suggest that this is the case for Chinese-English and Arabic-English. They train lexicalized reordering models (Tillmann, 2004; Koehn et al, 2007; Galley and Manning, 2008) on different but high quality parallel training data with specific genres. Their results show that the estimates of reordering parameters are significantly different between the corpora (e.g. the reordering probabilities estimated from News bilingual training data are different from the ones that are estimated from Legal bilingual data). It is therefore not so surprising that domain adaptation can help phrase-based SMT systems to improve reordering for English-Chinese as in (Chen et al, 2013a).

However, it is unlikely that this happens for all language pairs. We take English-Spanish as an example. Cuong and Sima'an (2014a) train different lexicalized reordering models on a somewhat similar scenario with News parallel training data, including four sub-corpora: EuroParl, Common Crawl Corpus, UN Corpus, News Commentary. They show that adapting reordering models for a new domain of *Consumer and Industrial Electronics* contributes a minor translation improvement for this domain.

As a side note, it is likely the case that dialect contributes to the reordering behavior (e.g. for Chinese (Chen et al, 2013a) and Egyptian Arabic (Jeblee et al, 2014)). Domain adaptation with respect to this aspect (e.g. training lexicalized reordering models on different dialect bilingual training data) therefore might contribute reordering improvements.

4.3 Optimization

Domain mismatch between held-out development and test data is also an important source of errors. This is widely observed in many studies, e.g. (Pecina et al, 2012; Nikoulina et al, 2012). We here show a qualitative example in Table 2. Specifically, we first train a phrase-based SMT system for English-German on a large dataset combined from multiple resources including EuroParl, Common Crawl Corpus and News Commentary. We then apply the system to a new domain of "*Legal Service*", but with three different scenarios for system optimization:

- We optimize the system on an in-domain (*Legal*) held-out development set with $2K$ sentence pairs.
- We optimize the system on a mixed-domain held-out development set with $8K$ sentence pairs from a combination of different domains: The in-domain *Legal* held-out development set itself, plus three different held-out development sets of *Software*, *Hardware* and *Professional & Business Services*.
- We optimize the system on another mixed-domain held-out development set with $6K$ sentence pairs of *Software*, *Hardware* and *Professional & Business Services* in the third setting. This is the mixed-domain held-out development set in the second setting, but excludes the in-domain development set part.

Note that there is no prior knowledge about the domain's provenance of the mixed-domain held-out development set in the second and third setting. Table 2 presents the translation performance of the phrase-based SMT system with respect to the different tuning scenarios.

It clearly indicates that moving to a new domain without having an in-domain held-out development set for system optimization can degrade the translation quality of a phrase-based SMT system. Note that our comparison may favor mixed-domain tuning scenarios: The mixed-domain held-out development sets are at least three times larger than the in-domain set, which presumably improves system optimization. In practice, the degradation of translation quality may be much more substantial, especially in an setting where the desired task is different from the held-out development set (e.g. Subtitles, Search Queries).

5 Domain adaptation: a general picture

A typical phrase-based SMT system contains various components, such as word alignment, language, translation and reordering models. This distinguishes SMT from most other Natural Language Processing tasks, and makes application of standard domain adaptation methods less straightforward.

In general, the most popular approach to domain adaptation for SMT is to induce domain-focused translation statistics from seed in-domain data. Domain-focused translation statistics are typically domain-specific phrase translation probability distributions, lexical weight and reordering probabilities. In the end, we could combine them together with the baseline domain-confused translation features, or even replace the baseline features. This results in a statistical translation framework with a combination of multiple (sub-)models for translation. Figure 4 provides an illustration of the de-facto standard approach to domain adaptation for SMT.

Implementing such a framework, however, is not trivial. Two main technical challenges are as follows:

- The induction of domain-focused translation statistics: specific prior knowledge (e.g. in-domain bilingual corpora, comparable corpora, monolingual

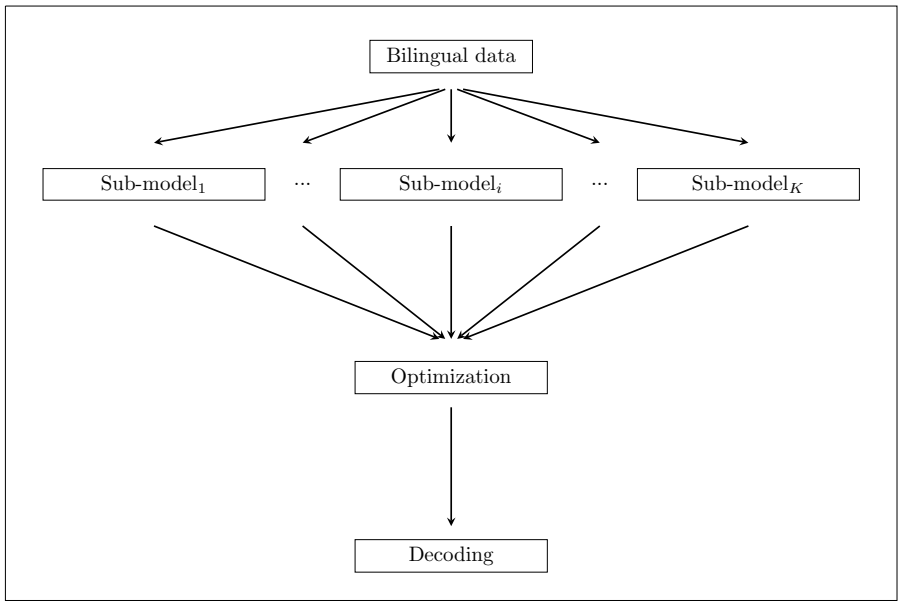

Fig. 4 Statistical translation framework with a combination of multiple K submodels for translation.

corpora) requires a different model for inducing domain-focused translation statistics. Section 6 provides a systematic overview of previous approaches to the problem.
– The combination of multiple (sub-)models for translation: the main object is a combination model tailored to high dimensional feature spaces, which is surprisingly hard to achieve. Section 7 reviews different combination models for adaptation.

Beside the two main research lines, previous work also considers other adaptation scenarios. This survey covers several adaptation trends (Section 8). We first review the induction of domain-focused sparse features and word alignment probabilities (Section 8.1 and Section 8.2). We also show how an existing system can be adapted to multiple specific domains at the same time (Section 8.3). Another scenario will be covered is applying an SMT system to web search queries (Section 8.4). We also discuss how web-based translation services can be improved when domain of a new request is not a priori known (Section 8.5 and Section 8.6).

6 Domain-specific translation induction for SMT

We start with induction with in-domain parallel data, and continue with comparable and monolingual corpora. We also discuss the induction with domain's provenance, which is special in the way that we are provided a large corpus con-

sisting of different domain-specific subcorpora that are not necessarily strictly related to the desired task.

6.1 Induction with in-domain parallel data

In many studies, a seed in-domain parallel corpus ($\mathcal{C}_{IN}$) exemplifying the target translation task is used as a form of prior knowledge for domain adaptation for SMT. The data, however, is very small compared with a mixed of domains corpus $\mathcal{C}_{OUT}$. The main goal of translation induction with in-domain parallel corpora is inducing a phrase-based model from $\mathcal{C}_{OUT}$ for adaptation. We now review the two most popular approaches to domain adaptation in this scenario: Instance weighting and Data selection.

Instance weighting
Instance weighting is perhaps the most effective approach to learning domain-focused translation statistics. To give some intuition about how instance weighting addresses the problem, in this general exposition we introduce a *latent domain variable* z to mark if a phrase is in-domain (z_1) or out-of-domain (z_0). With the introduction of the latent variable, we expect to extend the translation tables in phrase-based models from domain-confused $P(\tilde{e}|\ \tilde{f})$ to domain-focused by conditioning them on z, i.e. $P(\tilde{e}|\ \tilde{f},\ z)$. Note how $P(\tilde{e}|\ \tilde{f},\ z)$ contains $P(\tilde{e}|\ \tilde{f})$ as special case as follows:

$$P(\tilde{e}|\ \tilde{f},\ z) = \frac{P(\tilde{e}|\ \tilde{f})P(z|\ \tilde{e},\ \tilde{f})}{\sum_{\tilde{e}'} P(\tilde{e}'|\ \tilde{f})P(z|\ \tilde{e}',\ \tilde{f})}. \tag{14}$$

Here $P(z|\ \tilde{e},\ \tilde{f})$ is viewed as a latent phrase-relevance model, i.e. the probability that a phrase pair is in- (z_1) or out-domain (z_0). In the end, the adaptation can be performed by replacing the domain-confused tables, $P(\tilde{e}|\ \tilde{f})$, with the in-domain-focused ones, $P(\tilde{e}|\ \tilde{f},\ z_1)$, or simply by using these domain-focused model as additional feature for the baseline phrase-based SMT system.

From Eq 14, the main challenge of inducing $P(\tilde{e}|\ \tilde{f},\ z)$ then is inducing latent phrase-relevance model $P(z|\ \tilde{e},\ \tilde{f})$. Following Matsoukas et al (2009), a fairly large body of work on domain adaptation for SMT embeds $P(z|\ \tilde{e},\ \tilde{f})$ in an asymmetric sentence level model $P(z|\ \mathbf{e},\ \mathbf{f})$ for sentence pairs $\langle \mathbf{e}, \mathbf{f} \rangle$. Specifically, the estimation of $P(z|\ \tilde{e},\ \tilde{f})$ for phrases $\tilde{e}$ and $\tilde{f}$ can be simplified by computing $P(z|\ \mathbf{e},\ \mathbf{f})$ for sentence pairs $\langle \mathbf{e}, \mathbf{f} \rangle$ as follows:

$$P(z|\ \tilde{e},\ \tilde{f}) = \frac{\sum_{\mathbf{e},\ \mathbf{f}} P(z|\ \mathbf{e},\ \mathbf{f})\ c(\tilde{e};\ \mathbf{e})\ c(\tilde{f};\ \mathbf{f})}{\sum_{z' \in \{z_1,\ z_0\}} \sum_{\mathbf{e},\ \mathbf{f}} P(z'|\ \mathbf{e},\ \mathbf{f})\ c(\tilde{e};\ \mathbf{e})\ c(\tilde{f};\ \mathbf{f})}. \tag{15}$$

Here, $c(\tilde{e},\ \mathbf{e})$ and $c(\tilde{f},\ \mathbf{f})$ are the count of phrases $\tilde{e}$ and $\tilde{f}$ in sentence pairs $\langle \mathbf{e},\ \mathbf{f} \rangle$ in the training corpus.

But how to learn the asymmetric sentence level model? A simple and straightforward way, which is proposed by (Cuong and Sima'an, 2014a), is

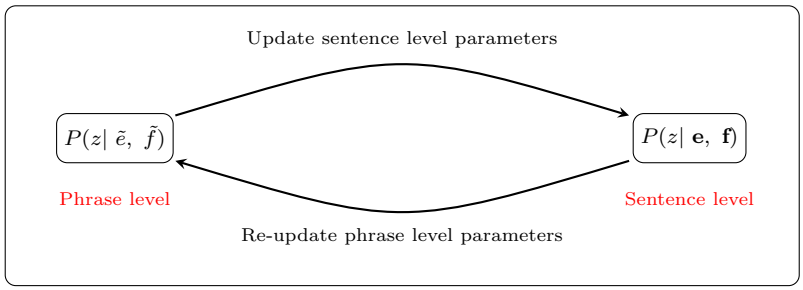

Fig. 5 The EM-based training algorithm for learning $P(z|\ \tilde{e},\ \tilde{f})$ and $P(z|\ \mathbf{e},\ \mathbf{f})$ simultaneously.

to devise an EM algorithm for learning (Figure 5). At every iteration, in- or out-domain estimates provide full sentence pairs $\langle \mathbf{e},\ \mathbf{f} \rangle$ with probabilities $P(z|\ \mathbf{e},\ \mathbf{f})$. The latent phrase-relevance model parameters are then re-estimated using these expectations. Metaphorically, during each EM iteration the current in- or out-domain phrase pairs compete in *inviting* $\mathcal{C}_{OUT}$ sentence pairs to be in- or out-domain, which bring in new (weights for) in- and out-domain phrases.

Another approach is directly building a logistic weighting model for the asymmetric sentence level model. Specifically, a logistic weighting model maps a set of features $\phi(\mathbf{e},\ \mathbf{f})$ with the parameter vector $\mathbf{w}$ to a scalar weight in $(0, 1)$. There are numerous types of sentence level features that can be used, such as manual sub-corpus and genre membership, number of source and target token, and ratio of number of the tokens on both sides. Interestingly, the parameter vector $\mathbf{w}$ can be learned directly simultaneously with log-linear model weight parameters so as to optimize the translation accuracy on a held-out development set. The approach was first proposed by Matsoukas et al (2009).

An alternative approach to learning domain-focused translation statistics is directly building a discriminative model at phrase level. This approach is intuitively plausible: a sentence often contains a mixture of domains by itself. In the work of Foster et al (2010), the estimation of domain-focused phrase translation probabilities can be directly computed as follows:

$$P(\tilde{e}|\ \tilde{f},\ IN) = \frac{c_{\mathbf{w}}(\tilde{e},\ \tilde{f})}{\sum_{\tilde{e}'} c_{\mathbf{w}}(\tilde{e}',\ \tilde{f})}, \tag{16}$$

where modified count $c_{\mathbf{w}}(\tilde{e},\ \tilde{f})$ is computed as follows:

$$c_{\mathbf{w}}(\tilde{e},\ \tilde{f}) = \frac{1}{1 + \exp(-\mathbf{w} \cdot \boldsymbol{\phi}(\tilde{e}, \tilde{f}))} c(\tilde{e},\ \tilde{f}). \tag{17}$$

Learning the weight parameters $\mathbf{w} = \{w_1, \ldots, w_K\}$ of K features for the logistic weighting model can be done using maximum likelihood or related criteria. More specifically, let us assume a held-out development set, in which each sentence $\langle \mathbf{e},\ \mathbf{f} \rangle$ contains a (multi-)set $\mathcal{A}(\mathbf{e},\ \mathbf{f})$ of extracted phrases $\langle \tilde{e},\ \tilde{f} \rangle$. The objective function is the maximization of the likelihood over $\mathcal{A}(\mathbf{e},\ \mathbf{f})$ for all parallel sentences $\langle \mathbf{e},\ \mathbf{f} \rangle$ in the development set, with respect to $\mathbf{w}$ as follows:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{\langle \mathbf{e},\ \mathbf{f} \rangle} \sum_{\langle \tilde{e},\ \tilde{f} \rangle \in \mathcal{A}(\mathbf{e},\mathbf{f})} \tilde{P}(\tilde{e}, \tilde{f}) \log P(\tilde{e} |\ \tilde{f},\ IN). \tag{18}$$

Here, note that $\tilde{P}(\tilde{e},\ \tilde{f})$ is computed from all phrase pairs extracted from the held-out development set. The optimization problem can be solved using the popular L-BFGS algorithm, as shown in Foster et al (2010). The algorithm requires computing the gradient $\frac{\partial P(\tilde{e} |\ \tilde{f},\ IN)}{\partial w_i}$, which is done as follows:

$$\frac{\partial P(\tilde{e} |\ \tilde{f},\ IN)}{\partial \lambda_i} = \frac{1}{P(\tilde{e} |\ \tilde{f},\ IN)} \Big[\frac{c_{\mathbf{w}_i}(\tilde{e},\ \tilde{f})}{\sum_{\tilde{f}'} c_{\mathbf{w}}(\tilde{e},\ \tilde{f}')} - \frac{c_{\mathbf{w}}(\tilde{e},\ \tilde{f}) \sum_{\tilde{f}'} c_{\mathbf{w}_i}(\tilde{e},\ \tilde{f}')}{(\sum_{\tilde{f}'} c_{\mathbf{w}}(\tilde{e},\ \tilde{f}'))^2} \Big]. \tag{19}$$

where:

$$c_{\mathbf{w}_i}(\tilde{e},\ \tilde{f}) = c_{\mathbf{w}}(\tilde{e},\ \tilde{f}) f_i(\tilde{e},\ \tilde{f}) \Big(\frac{\exp(-\mathbf{w} \cdot \boldsymbol{\phi}(\tilde{e},\ \tilde{f}))}{1 + \exp(-\mathbf{w} \cdot \boldsymbol{\phi}(\tilde{e},\ \tilde{f}))} \Big). \tag{20}$$

Both these different approaches of course have their own advantages and disadvantages. The EM-based approach strikes for the simplicity and thus it is much easier to implement. Meanwhile, using a discriminative model to learn relevance of sentence pairs and phrases in the parallel training data would perhaps be much more effective. The discriminative models, however, require feature engineering. They are also more difficult to implement. An empirical comparison of the approaches, however, is not thoroughly conducted yet in the literature, to the best of our knowledge.

Note that using the same algorithm we can also adapt all other core translation components in tandem, including lexical weight and lexicalized reordering models.

Data selection

Another approach to learning domain-focused translation statistics is selecting training data from a large corpus. Then, we can simply train a phrase-based SMT system on the selected data. Resulting translation statistics are presumably domain-focused. Data selection would naturally be less effective than instance weighting, as we strictly remove a lot of bilingual data that are (presumably) not relevant to a desired task. However, data selection has received considerable attention in the past years because of two main reasons:

– Large bilingual training data comes with a cost: training phrase-based SMT systems on large data is extremely expensive and time-consuming.

– A small percentage, but well-selected of the data often outperforms the full dataset for training a phrase-based SMT system (Axelrod et al, 2011; Mansour et al, 2011; Zhang and Chiang, 2014; Kirchhoff and Bilmes, 2014; Duh et al, 2013; Mansour and Ney, 2014; Cuong and Sima'an, 2014b).

Existing work can be roughly classified depending on what kind of information is used for selection. The most popular approach (Axelrod et al, 2011) selects sentence pairs using the cross-entropy difference between in- and out-of-domain language models (both source and target sides):

$$rank(\mathbf{f},\ \mathbf{e}) = \underbrace{\Big(H_{LM_{IN}}(\mathbf{f}) - H_{LM_{OUT}}(\mathbf{f})\Big)}_{source\ side} + \underbrace{\Big(H_{LM_{IN}}(\mathbf{e}) - H_{LM_{OUT}}(\mathbf{e})\Big)}_{target\ side}. \tag{21}$$

The cross-entropy is defined as

$$H_{LM}(\mathbf{f}) = -\frac{1}{m}\sum\nolimits_{i=1}^{m} \log P(f_i|\ f_1^{i-1}) \tag{22}$$

$$H_{LM}(\mathbf{e}) = -\frac{1}{l}\sum\nolimits_{i=1}^{l} \log P(e_i|\ e_1^{i-1}) \tag{23}$$

The method itself is a modification of the method proposed in Moore and Lewis (2010), which was introduced to address exactly the same problem we are discussing, but for only one side (i.e. monolingual data).

More recent approaches (Mansour et al, 2011; Mansour and Ney, 2014; Cuong and Sima'an, 2014b) use translation model information. The idea is intuitively plausible: in the translation context, often a source phrase has different translations in different domains, which cannot be distinguished with monolingual language models. But how much should data selection depend on bilingual vs. monolingual factors? Cuong and Sima'an (2014b) present a comprehensive study of the contribution of these factors, showing that they actually complement each other for data selection.

One of the most difficult problems in data selection is to jointly learn translation and language models. An EM-based learning algorithm was first proposed by Cuong and Sima'an (2014b) to address the problem. However, a joint bilingual neural network model proposed by Devlin et al (2014) might be a more powerful solution to the problem. Chen et al (2016) was the first study that deploys the bilingual neural network joint model to address the problem. In their work, promising data selection performance is observed.

As a side note, data selection complements with data reduction for SMT (Eck et al, 2005; Lewis and Eetemadi, 2013). Data reduction aims at reducing the size of data that is used for training, yet which at the same time has little impact on quality.

6.2 Induction with comparable corpora

Creating an in-domain dataset is extremely expensive in practice. A cheaper approach to domain adaptation for SMT is mining comparable corpora (Snover et al, 2008; Daumé and Jagarlamudi, 2011; Irvine et al, 2013b).

We now present two notable approaches as examples. The first approach is mining unseen words for an adaptation task (Daumé and Jagarlamudi, 2011). It extends the approach described in Haghighi et al (2008) to mining translations from comparable corpora. Learning bilingual lexicons from comparable corpora is obviously not an easy task (e.g. see (Koehn and Knight, 2002; Haghighi et al, 2008; Tamura et al, 2012)), and their mining technique is "bootstrapped" based on a bilingual dictionary that is created automatically from out-of-domain corpora. The output of the dictionary mining approach is normally a list of word pairs (source and target words), with corresponding scores representing the word translation degree. Perhaps surprisingly, a straightforward approach to incorporating the induced word pairs by having an additional feature of dictionary mining translation probability may not be helpful. A more effective way, as described in Daumé and Jagarlamudi (2011), is to having not only the feature of dictionary mining translation probability, but also an additional feature to mark if a phrase pair is seen in the source and target data or not.

The second approach, proposed by Irvine et al (2013b), directly recovers the joint probability distribution of source and target word pairs on a new domain. Specifically, let us assume we have access to a joint distribution $P_{OUT}(f,\ e)$ over source and target word pairs $\langle f,\ e\rangle$. The distribution is estimated from an out-of-domain corpus. Let $\tilde{P}(f)$ and $\tilde{P}(e)$ be the empirical marginal distributions estimated from comparable corpora (i.e. we extract raw word frequencies from the corpora). Irvine et al (2013b) cast the learning of the joint probability distribution of source and target word pairs on a new domain as a linear programming problem, as follows:

$$\hat{P}_{IN} = \underset{P_{IN}}{\operatorname{argmin}} \| \sum_{\langle f,\ e\rangle} P_{IN}(f,e)\ -\ P_{OUT}(f,e) \|_1, \tag{24}$$

subject to:

$$\sum_{\langle f,e\rangle} P_{IN}(f,e) = 1, \sum_{e} P_{IN}(f,e) = \tilde{P}(f), \sum_{f} P_{IN}(f) = \tilde{P}(e),\ \text{and}\ P_{IN}(f,e) \geq 0.$$

Here, l_1-norm ($\|\cdot\|_1$) is used to measure the distance between two distributions. Regularization terms are usually added into Eq. 24 so that the solution would be as sparse as possible. A linear programming solver can be used to learn $P_{IN}(f,\ e)$ from Eq. 24.

The method is perhaps one of the most elegant approaches to domain adaptation for SMT. It exploits cheap resources and shows significant improvement in translation quality on new domains.

6.3 Induction with monolingual data

Exploiting in-domain monolingual data is also an effective approach to domain adaptation for SMT. In general, synthetic bilingual data is first generated by using a phrase-based SMT system. Then, we can use the created data to induce domain-focused translation statistics (Schwenk, 2008; Wu et al, 2008; Bertoldi and Federico, 2009; Schwenk and Senellart, 2009). Empirical results show that having in-domain monolingual data could substantially improve the translation quality for a new domain, especially with in-domain monolingual data on the target side (Lambert et al, 2011).

Surprisingly, we can still get improvements from incorporating induced domain-focused translation features to the baseline, given that the baseline is already augmented with induced domain-focused language model features. As a side note, the adaptation of reordering model gives consistent but modest improvement in this scenario (Bertoldi and Federico, 2009; Schwenk, 2008).

6.4 Induction with monolingual data and meta-information

Beside generating synthetic bilingual data, are there any other ways of adapting translation models with monolingual corpora? There has been an intensive line of research that focuses on the translation model adaptation using topic models (Su et al, 2012; Eidelman et al, 2012; Gong et al, 2011; Hewavitharana et al, 2013; Hasler et al, 2014; Hu et al, 2014). Such studies interchangeably use the term "topic" and "domain".

Let us assume we are provided with an out-of-domain parallel corpus $\mathcal{C}_{OUT} = \{\mathcal{S}_{OUT},\ \mathcal{T}_{OUT}\}$, together with an in-domain monolingual corpus on source side $\mathcal{S}_{IN}$ only. Given the data, a general approach is building an adapted translation model in the following steps:

- Step 1: Estimating topic models (e.g. Probabilistic Latent Semantic Analysis (Hofmann, 1999), Latent Dirichlet Allocation (Blei et al, 2003), Hidden Topic Markov Models (Gruber et al, 2007)) at document level in monolingual corpora.
- Step 2: Estimating topic-specific translation models (i.e. conditioning the translation of phrase pairs on the topic information of source phrases).
- Step 3: Estimating topic posterior distributions of phrases.
- Step 4: Estimating phrase translation probabilities using predefined topic-specific translation models and topic posterior distributions of phrases.

More formally, let us use $P(z_{\mathbf{f}_{IN}}|\ \mathbf{f})$ and $P(z_{\mathbf{f}_{OUT}}|\ \mathbf{f})$ to indicate how a sentence $\mathbf{f}$ expresses a specific source-side topic in in- and out-domain monolingual corpus. The sentence-topic distributions are provided by topic models (Step 1).

Let us use $P(\tilde{e}|\ \tilde{f}, z_{\tilde{f}_{OUT}})$ to indicate the probability of translating phrase $\tilde{f}$ to phrase $\tilde{e}$ given the source-side topic $z_{\tilde{f}_{OUT}}$. The topic-specific translation

models are estimated as follows (Step 2):

$$P(\tilde{e}|\ \tilde{f},\ z_{\tilde{f}_{OUT}}) = \frac{\sum_{\mathbf{e},\ \mathbf{f} \in\ \mathcal{C}_{OUT}} P(z_{\tilde{f}_{OUT}}|\ \mathbf{f})\ c(\tilde{f};\ \mathbf{f})\ c(\tilde{e};\ \mathbf{e})}{\sum_{\tilde{e}'} \sum_{\mathbf{e},\ \mathbf{f} \in\ \mathcal{C}_{OUT}} P(z_{\tilde{f}_{OUT}}|\ \mathbf{f})\ c(\tilde{f};\ \mathbf{f})\ c(\tilde{e}';\ \mathbf{e})}. \quad (25)$$

Let us use $P(z_{\tilde{f}_{IN}}|\ \tilde{f})$ and $P(z_{\tilde{f}_{OUT}}|\ \tilde{f})$ to denote the phrase-topic distributions. The distributions can be computed as follows (Step 3):[4]

$$P(z_{\tilde{f}_{IN}}|\ \tilde{f}) = \frac{\sum_{\mathbf{f} \in\ \mathcal{S}_{IN}} P(z_{\tilde{f}_{IN}}|\ \mathbf{f})\ c(\tilde{f};\ \mathbf{f})}{\sum_{z'_{\tilde{f}_{IN}}} \sum_{\mathbf{f} \in\ \mathcal{S}_{IN}} P(z'_{\tilde{f}_{OUT}}|\ \mathbf{f})\ c(\tilde{f};\ \mathbf{f})}. \quad (26)$$

$$P(z_{\tilde{f}_{OUT}}|\ \tilde{f}) = \frac{\sum_{\mathbf{f} \in\ \mathcal{S}_{OUT}} P(z_{\tilde{f}_{OUT}}|\ \mathbf{f})\ c(\tilde{f};\ \mathbf{f})}{\sum_{z'_{\tilde{f}_{OUT}}} \sum_{\mathbf{f} \in\ \mathcal{C}_{OUT}} P(z'_{\tilde{f}_{OUT}}|\ \mathbf{f})\ c(\tilde{f};\ \mathbf{f})}. \quad (27)$$

Finally, phrase translation probabilities can be computed as follows (Step 4):

$$P(\tilde{e}|\ \tilde{f}) = \sum\nolimits_{z_{\tilde{f}_{IN}}} \sum\nolimits_{z_{\tilde{f}_{OUT}}} P(\tilde{e}|\ \tilde{f},\ z_{\tilde{f}_{OUT}}) P(z_{\tilde{f}_{OUT}}|\ z_{\tilde{f}_{IN}}) P(z_{\tilde{f}_{IN}}|\ \tilde{f}), \quad (28)$$

where topic mapping probability distribution $P(z_{\tilde{f}_{OUT}}|z_{\tilde{f}_{IN}})$ can be computed as:[5]

$$P(z_{\tilde{f}_{OUT}}|z_{\tilde{f}_{IN}}) = \sum\nolimits_{\tilde{f} \in \mathcal{S}_{IN}\ \cap\ \mathcal{S}_{OUT}} P_{IN}(z_{\tilde{f}_{IN}}|\ \tilde{f}) P_{OUT}(z_{\tilde{f}_{OUT}}|\ \tilde{f}). \quad (29)$$

The estimate of $P(\tilde{e}|\ \tilde{f})$ as in Eq. 28 can be used to replace the domain-confused translation probability. It can also simply serve as an additional feature to the baseline.

In practice, it is also possible that instead of having only source side $\mathcal{S}_{IN}$ of monolingual data, we are provided with an in-domain parallel corpus $\mathcal{C}_{IN} = \{\mathcal{S}_{IN},\ \mathcal{T}_{IN}\}$. In that case, bilingual topic inference should be preferred to monolingual topic inference (Mimno et al, 2009; Hasler et al, 2014; Hu et al, 2014).

Using topic models for domain adaptation for SMT provides an effective way of quantifying the effect of the topical context information on translation selection. Using the same approach, we can adapt all other core translation components in tandem, including lexical weight and lexicalized reordering models.

Meanwhile, the model has a potential drawback: most of parallel corpora lack the annotation of document boundaries. Of course, a single sentence can be considered as a short pseudo-document, but it is questionable whether such a corpus with short pseudo-documents is topic-model "friendly" (Tang et al, 2014).

[4] In Su et al (2012), an interpolation model is computed for $P_{IN}(z_{\tilde{f}_{IN}}|\ \tilde{f})$, which is decomposed into the topic posterior distribution at word level for smoothing.

[5] Joint inference of topic models on a concatenation of $\mathcal{S}_{IN}$ and $\mathcal{S}_{OUT}$ would drop the requirement of computing the topic mapping probability distribution (e.g. see Gong et al (2011) and Hewavitharana et al (2013)). An empirical comparison of the approaches, however, is not thoroughly conducted yet in the literature, to the best of our knowledge.

6.5 Induction with domain's provenance

In practice, there are adaptation scenarios where we are provided with a large corpus consisting of different domain-specific subcorpora (the subcorpora are manually grouped/annotated). The subcorpora are not necessarily strictly related to the desired task. In that scenario, it is still very useful to condition the lexical weighting features on provenance (Chiang et al, 2011). In the end, we can simply optimize the system with different types of domain-focused translation statistics on an in-domain held-out development set.

Another simple and elegant approach is to use a vector space model. Specifically, let us assume we are provided with a corpus consisting of N different domain-specific subcorpora. First, we create a vector profile for every phrase pair extracted from the training data as follows:

$$V_{training}(\tilde{f},\ \tilde{e}) = \Big[w_1(\tilde{f},\ \tilde{e}), \ldots, w_N(\tilde{f},\ \tilde{e})\Big] \tag{30}$$

Another vector profile is created for every phrase pair extracted from the in-domain held-out development set as:

$$V_{dev}(\tilde{f},\ \tilde{e}) = \Big[w_1(\tilde{f},\ \tilde{e}), \ldots, w_N(\tilde{f},\ \tilde{e})\Big] \tag{31}$$

In principle, each element of the vector $w(\tilde{f},\ \tilde{e})$ can be simply the count of a phrase pair. A better approach, as proposed by Chen et al (2013b), is adapting standard $tf - idf$ statistics - a standard technique in information retrieval.

Then, we simply use the similarity score between these two types of vectors as additional feature functions (e.g. the Bhattacharyya distance (Bhattacharyya, 1946), the Kullback-Leibler distance (Kullback and Leibler, 1951), and the cosine distance). These feature functions reward phrase pairs that are relevant to the desired task.

The vector space model approach was first proposed by Chen et al (2013b). It is a very effective adaptation technique for SMT. However, domain's provenance is not always available in practice. Despite topic models can automatically provide meta-information, experiments in the setting show a modest improvement (e.g. see Hewavitharana et al (2013)).

7 Model combination for adaptation

Domain-focused translation statistics, once induced, need to be combined together in an appropriate way. The main desire is to have a combination model tailored to high dimensional feature spaces.

7.1 Log-linear Mixture

Log-linear translation model mixtures (Birch et al, 2007; Koehn and Schroeder, 2007) are of the form:

$$\phi_{TM}(\mathbf{e},\ \mathbf{f}) = \lambda \sum_{i=1}^{n} \log P(\tilde{e}_i|\ \tilde{f}_{a_i}, IN) + (1-\lambda) \sum_{i=1}^{n} \log P(\tilde{e}_i|\ \tilde{f}_{a_i},\ OUT). \tag{32}$$

Here, $P(\tilde{e}_i|\ \tilde{f}_{a_i}, IN)$ and $P(\tilde{e}_i|\ \tilde{f}_{a_i}, IN)$ represent different types of domain-focused translation statistics with respect to IN and OUT. As in Eq. 32, they can be added to the baseline as additional features. There is also no further effort needed for training: the respective weights are set with any weight optimization method (e.g. MERT, MIRA, PRO).

The implementation of log-linear translation mixture model for adaptation can be slightly different in practice. It is common to leave the decoder "*as is*" Razmara et al (2012), but it is also possible to put constraints on hypotheses generated by the decoder (Birch et al, 2007; Koehn and Schroeder, 2007). For instance, the decoder may only generate the hypotheses that are contained in both translation tables (both in-domain and out-of-domain translation table). The decoder may also generate the hypotheses that are contained in each of the tables. An empirical comparison of the implementations, however, is not thoroughly conducted yet in the literature, to the best of our knowledge.

The model has two potential drawbacks:

- In practice, it is common to have many submodels. This leads to significantly longer search and potentially more search errors. This also makes system optimization even more challenging. It is not uncommon for such a log-linear mixture model to perform significantly worse than for system trained on a concatenation of all the data (Wäschle and Riezler, 2012; Sennrich, 2012).
- Having high dimensional feature spaces requires a much larger held-out development set for system optimization (Waite and Byrne, 2015). This is unrealistic in practice, as in-domain data is very expensive to annotate.

7.2 Linear Mixture

Linear translation model mixtures are of the form:

$$\phi_{TM}(\mathbf{e},\mathbf{f}) = \sum_{i=1}^{n} \log \Big(\lambda P(\tilde{e}_i|\ \tilde{f}_{a_i},\ IN) + (1-\lambda) P(\tilde{e}_i|\ \tilde{f}_{a_i},\ OUT) \Big) \tag{33}$$

An alternative form of linear combination is a maximum a posteriori (MAP) combination as follows:

$$\phi_{TM}(\mathbf{e},\mathbf{f}) = \sum_{i=1}^{n} \log \left(\frac{c_{IN}(\tilde{e}_i,\ \tilde{f}_{a_i}) + \lambda P(\tilde{e}_i|\ \tilde{f}_{a_i}, OUT)}{\sum_{\tilde{e}'} c_{IN}(\tilde{e}',\ \tilde{f}_{a_i}) + \lambda} \right). \tag{34}$$

This model was first proposed by Foster and Kuhn (2007).

Training the model is not straightforward. A desire is to directly optimize the weights of the baseline system $\mathbf{w} = \{w_1, \ldots, w_M\}$, and interpolation weight λ directly for BLEU. This is possible (Haddow, 2013; Foster et al, 2013), but very challenging to implement.[6] In practice, the most common approach is performing system optimization with a two-step procedure as follows:

- First, we learn the interpolation weight by maximum likelihood or related criteria.
- We held the interpolation weight as constant, and optimize the log-linear weights as normal with any optimization method (e.g. MERT, MIRA, PRO)

By isolating the task of learning log-linear weights, the problem of learning the interpolation weight is not hard (Foster et al, 2010; Sennrich, 2012). Specifically, let us assume a held-out development set, in which each sentence $\langle \mathbf{e},\ \mathbf{f} \rangle$ contains a (multi-)set $\mathcal{A}(\mathbf{e},\ \mathbf{f})$ of extracted phrases $\langle \tilde{e}, \tilde{f} \rangle$. The objective function is the maximization of the likelihood over $\mathcal{A}(\mathbf{e},\ \mathbf{f})$ for all pairs $\langle \mathbf{e},\ \mathbf{f} \rangle$, with respect to λ as follows:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} \sum_{\langle \mathbf{e},\ \mathbf{f} \rangle} \sum_{\langle \tilde{e},\ \tilde{f} \rangle \in \mathcal{A}(\mathbf{e},\mathbf{f})} \tilde{P}(\tilde{e}, \tilde{f}) \log \Big(\lambda P(\tilde{e}|\ \tilde{f}, IN) + (1-\lambda) P(\tilde{e}|\ \tilde{f}, OUT) \Big). \tag{35}$$

In case of using MAP, the objective function of training is as follows:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} \sum_{\langle \mathbf{e},\ \mathbf{f} \rangle} \sum_{\langle \tilde{e},\ \tilde{f} \rangle \in \mathcal{A}(\mathbf{e},\mathbf{f})} \tilde{P}(\tilde{e}, \tilde{f}) \log \frac{c_{IN}(\tilde{e},\ \tilde{f}) + \lambda P(\tilde{e}|\ \tilde{f}, OUT)}{\sum_{\tilde{e}'} c_{IN}(\tilde{e},\ \tilde{f}) + \lambda}. \tag{36}$$

Note that $\tilde{P}(\tilde{e},\ \tilde{f})$ in both cases is computed from all phrase pairs extracted from the held-out development set.

Since the objective function is convex, the optimization can be done efficiently with EM (Carpuat et al, 2014) or Limited-memory BFGS algorithm (Sennrich, 2012).[7] Both algorithms require computing the gradient $\frac{\partial}{\partial \lambda}$. The gradient is easy to compute in the first case as follows:

$$\frac{\partial}{\partial \lambda} = \left[\frac{P(\tilde{e}|\ \tilde{f}, IN) - P(\tilde{e}|\ \tilde{f}, OUT)}{\lambda P(\tilde{e}|\ \tilde{f}, IN) + (1-\lambda) P(\tilde{e}|\ \tilde{f}, OUT)} \right] \tag{37}$$

In case of using a MAP, the gradient is slightly different as follows:

$$\frac{\partial}{\partial \lambda} = \frac{-\sum_{\tilde{e}'} c_{IN}(\tilde{e}',\ \tilde{f})}{\left(\sum_{\tilde{e}'} c_{IN}(\tilde{e}',\ \tilde{f}) + \lambda\right)^2} \left[\frac{P(\tilde{e}|\ \tilde{f}, IN) - P(\tilde{e}|\ \tilde{f}, OUT)}{\frac{c_{IN}(\tilde{e},\ \tilde{f}) + \lambda \bar{P}(\tilde{e}|\ \tilde{f}, OUT)}{\sum_{\tilde{e}'} c_{IN}(\tilde{e}',\ \tilde{f}) + \lambda}} \right] \tag{38}$$

[6] There has not been any attempt at such an implementation for combining multiple submodels, as far as we are aware.

[7] The Expectation Maximization algorithm often gives a more efficient and stable performance in practice (e.g. see Razmara et al (2012)).

A linear translation model is perhaps the most common combination model for adaptation. Compared with the log-linear translation model, it often works better with high dimensional feature spaces. The model, however, has two potential drawbacks:

- The maximum likelihood or related criteria may not correlate well with translation accuracy. It is not uncommon that assigning optimized weights underperforms uniform weights.
- The performance would likely be suffered from combining too many submodels (e.g. more than 10 submodels), leaving an open question of designing a combination model tailored to very high dimensional feature spaces.

7.3 Fill-up

A very simple approach that provides a competitive performance to log-linear and linear translation model mixtures is Fill-up. The idea of Fill-up was first proposed by Besling and Meier (1995) for addressing the problem of language model adaptation for speech recognition. It was first introduced in SMT by Nakov (2008), and first used in domain adaptation for SMT in the work of Bisazza et al (2011).

Let us assume we have two translation tables T_{IN} and T_{OUT}, with their corresponding phrase translation probabilities $P(\tilde{e}|\ \tilde{f},\ IN)$ and $P(\tilde{e}|\ \tilde{f},\ OUT)$ respectively. A Fill-up table T_{FILLUP} is defined as:

$$\forall(\tilde{f},\tilde{e}) \in T_{IN}\ \cup\ T_{OUT}: \tag{39}$$
$$T_{FILLUP}(\tilde{f},\ \tilde{e}) = \begin{cases} \{P(\tilde{e}|\ \tilde{f},\ IN), \exp(0)\} & \text{if } (\tilde{f},\ \tilde{e}) \in T_{IN} \\ \{P(\tilde{e}|\ \tilde{f},\ OUT), \exp(1)\} & \text{otherwise.} \end{cases}$$

Here, the entries of T_{FILLUP} correspond to the union of the two phrase tables, in which we consider T_{IN} as the more reliable source and use it whenever possible. The exponential function (i.e. $\exp(0)$ and $\exp(1)$) is to mark if a phrase pair is in-domain (T_{IN}) or out-domain (T_{OUT}).

Simplicity is perhaps the main advantage of Fill-up. The model, however, has two potential drawbacks:

- It remains unclear whether the approach is able to scale to many submodels. Such an empirical evaluation is not thoroughly conducted yet in the literature, to the best of our knowledge.
- Translation probabilities in T_{FILLUP} do not form a full probability distribution. This is potentially problematic: interactions between features can be complex and log-linear models may not be able to handle the interactions.

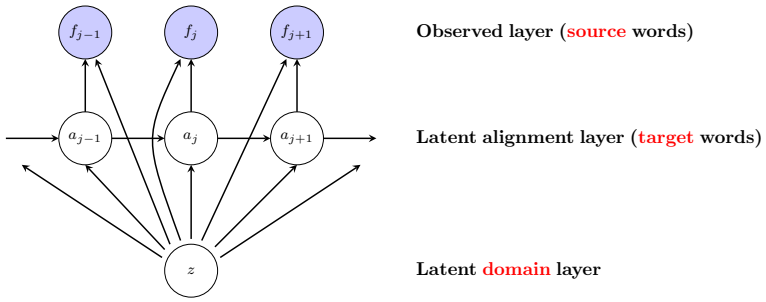

Fig. 6 Latent domain HMM alignment model. An additional latent layer representing domains has been conditioned on by both the rest two layers.

8 Other trends in domain adaptation

This survey covers several other adaptation trends. We first review the induction of domain-focused sparse features and word alignment probabilities (Section 8.1 and Section 8.2). We also show how an existing system can be adapted to multiple specific domains at the same time (Section 8.3). Another scenario is applying an SMT system to web search queries (Section 8.4). We also discuss how web-based translation services can be improved when domain of a new request is not a priori known (Section 8.5 and Section 8.6).

8.1 Adaptation with sparse features

Having in-domain sparse feature functions is particularly useful when applying a phrase-based SMT system to new domains (Bertoldi and Federico, 2009; Hasler et al, 2012; Green et al, 2013, 2014). This is because sparse features allow for more flexibility than dense features. This, however, is at the risk of raising the difficulty of the optimization. Applying cross-validation techniques (e.g. jackknife training (Hasler et al, 2012)) is often very useful to avoid overfitting.

8.2 Domain adaptation for word alignment

Word alignment models, like any statistical models, would presumably suffer significantly from lacking in-domain data for training. There is some evidence supporting this. Hua et al (2005) train different alignment models independently on different domain-specific subcorpora. In the end, they show that an interpolation of the alignment models improves word alignment accuracy.

Similar findings are reported in Gao et al (2011) and Duh et al (2010). Gao et al (2011) show that an interpolation of domain-specific and general-domain alignment model improves translation accuracy. Duh et al (2010) suggest that training a phrase-based SMT system might benefit from using a simple trick

as follows. They first train statistical alignment models on a concatenation of both in-domain and a much larger out-of-domain dataset. Then, they exclude out-of-domain data during phrase extraction.

As a side note, Shah et al (2010) show that it would benefit from training word alignment with weighting sentence pairs according to their relevance to a new domain.

Recently, Cuong and Sima'an (2015) provide an in-depth study of domain adaptation for word alignment. They focus on the insensitivity of existing word alignment models to domain differences, which often yields suboptimal results on heterogeneous corpora (e.g. EuroParl, Common Crawl Corpus, UN Corpus, News Commentary). A latent domain word alignment model is proposed, which explicitly incorporates latent domain information in learning domain-focused lexical and alignment statistics. Figure 6 presents such a case with a latent domain HMM alignment model. Cuong and Sima'an (2015) train the model on a heterogeneous corpus, using a small number of seed samples from different domains. Their experiments show that the derived domain-focused statistics, once combined together, produce significant improvements both in word alignment and translation.

8.3 Multi-domain adaptation for SMT using multi-task learning

A common scenario in practice is adapting an existing system to multiple domain-specific tasks at the same time. This is obviously challenging.

The main approach is optimizing an SMT system in the way that exploits commonalities shared among different tasks (Wäschle and Riezler, 2012). More formally, let us use $\{\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_K\}$ to denote a set of model parameters with respect to K different domains. The commonalities shared among different tasks are modeled as follows:

$$\mathbf{w}_{AVG} = \frac{1}{K} \sum\nolimits_{d=1}^{K} \mathbf{w}_d. \tag{40}$$

In the end, the goal is to learn model parameters that maximize the objective function as follows:

$$\{\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_K\} = \underset{\mathbf{w}_1,\ldots,\mathbf{w}_K}{\operatorname{argmin}} \sum\nolimits_{d=1}^{K} loss_d(\mathbf{w}_d) + \lambda \sum\nolimits_{d=1}^{K} \|\mathbf{w}_d - \mathbf{w}_{AVG}\|_1. \tag{41}$$

Here, the parameter λ controls the influence of the regularization, which trades off between task-specific parameter vectors and their distance to the average. Meanwhile, we use $loss_d(\mathbf{w}_d)$ to represent a translation loss function on the held-out development set from task d. The optimization problem can be solved using gradient descent optimization with l_1-regularization (Tsuruoka et al, 2009; Wäschle and Riezler, 2012).

While this approach is intuitively plausible, it gives a modest translation improvement Wäschle and Riezler (2012). Other approaches are also proposed in the literature (e.g. Cui et al (2013); Simianer et al (2012)), showing a potentially more promising performance.

8.4 Cross Lingual Information Retrieval (CLIR)

A practical real-world problem is translating web search queries into several target languages, so that a search engine can return search results in the corresponding languages. The quality of a translation component thus plays a crucial role. The problem, however, is particularly difficult because of three specific reasons:

- Translation quality degrades substantially when applying a phrase-based SMT system to a domain-specific task. This is particularly true for search queries, due to their unique characteristics: search queries are very short (just a couple of words per query) and the word order would be different to a typical sentence in natural language.
- Second, a phrase-based SMT system is usually trained to optimize the quality of the translation, which is not necessarily correlated with the retrieval quality (especially for the short queries) (Nikoulina et al, 2012; Kettunen, 2009). For example, the word order which is crucial for translation quality is often ignored by Information Retrieval models. In contrast, retrieval systems often use bag-of-word representations in document scoring models, and queries are rarely grammatical natural language.
- Finally, there are only a few tiny corpora of parallel queries (e.g. CLEF tracks) that can be obtained.

A very simple, yet effective approach to improving adaptation for CLIR is reranking the N-best translation candidates generated by a baseline system (Nikoulina et al, 2012). Note that a re-ranker should be optimized to maximize a retrieval metric rather than translation accuracy. Putting constraints on hypotheses generated by the decoder is also another approach to improving adaptation for CLIR (Dong et al, 2014; Hieber and Riezler, 2015). While the later approach may be more efficient, such an implementation is obviously far more complicated.

8.5 Cache-based adaptive models for translation adaptation

A common scenario in practice, particularly for web-based translation services such as Bing Translator and Google Translate, is that translation requests are *unknown* in their domain. A common approach is exploiting two general phenomenons in natural language and translation:

- Repetition and recency effects of words: many words, especially content words, are repeated in close context.
- Consistency of translations: the translation of content words is consistent given a specific context.

The two phenomenons provide us with a natural way to perform a fully unsupervised domain adaptation on a new domain: a phrase-based SMT system performs the translation for a sentence by not only considering the sentence

itself, but also taking the translation history of recent input sentences into account.

Accounting for the phenomenons in the translation is fairly simple, using a cache-based adaptive model (Kuhn and De Mori, 1990). More specifically, Tiedemann (2010) develops two cache-based adaptive models as follows:

Cache-based adaptive language model: Tiedemann (2010) uses a dynamic cache-based adaptive language models in the form of linear mixtures as we discuss below:

$$P(e_n|\ e_{n-k},\ \ldots,\ e_{n-1}) = (1\ -\ \lambda)P(e_n|\ e_{n-k},\ \ldots,\ e_{n-1},\ OUT) + \lambda P(e_n|\ e_{n-k},\ \ldots,\ e_{n-1},\ CACHE) \quad (42)$$

Here, the cache stores the best translation hypotheses of previous sentences. Of course the size of the cache would be very small (e.g. 100-5000 words). The value of interpolation weight λ can be set manually. The EM algorithm can also be used to learn the weight automatically.

Implementing the model as a simple unigram model is a good option, but a better solution in practice would be introducing a decay factor in the estimation of cache probabilities as follows:

$$P(e_n|\ e_{n-k},\ \ldots,\ e_{n-1},\ CACHE) \propto \sum_{i=n-k}^{n-1} \delta(e_n\ =\ e_i) \exp\Big(-\alpha(n-i)\Big) \quad (43)$$

This approach was first introduced by Clarkson and Robinson (1997). Here, δ is the Kronecker delta function. The decay rate α is normally set to a very small value (e.g. 0.005 as in Clarkson and Robinson (1997)).

Cache-based adaptive translation model: (Tiedemann, 2010) develops a cache-based adaptive translation model in a similar manner, using a decay factor to compute translation model scores from the cache as follows:

$$P(\tilde{e}_n|\ \tilde{f}_n,\ CACHE) \propto \sum_{i=1}^{K} \delta\Big(\langle \tilde{e}_n,\ \tilde{f}_n\rangle\ =\ \langle \tilde{e}_i,\ \tilde{f}_i\rangle\Big) \exp\Big(-\alpha i\Big) \quad (44)$$

The cache-based adaptive models can be integrated into a phrase-based SMT system in a straightforward manner: both can be used to replace the language and translation models, or to serve as additional feature functions within a log-linear model. In the end, the decoder is forced to prefer identical translations for repeated terms.

While using cache-based adaptive models is an elegant approach, the adaptation effect is rather modest (Tiedemann, 2010). The effect is also not robust: it is not uncommon that an augmented SMT system produces a rather sub-optimal translation. There are two potential reasons for this:

– First, it would be risky to assume that previous translation hypotheses are good enough to be cached (i.e. the risk of error propagation (Tiedemann, 2010)).
– Second, using the translation of initial sentences in the input stream may not be so beneficial.

Potential solutions to these problems are quite straightforward (Gong et al, 2011; Louis and Webber, 2014). For instance, in the work of Gong et al (2011), the cache stores similar target sentence pairs in the bilingual training data to the translation hypotheses, instead of the translation hypotheses by themselves. As a side note, other types of caches can be developed to improve adaptation, e.g. caching not only phrase pairs but also topic caches, as in Gong et al (2011).

8.6 Rewarding domain invariance for adaptation

When the target domain is unknown at training time, the system could be also trained to make safer choices, preferring translations which are likely to work across different domains. For example, when translating from English to Russian, the most natural translation for the word 'code' would be highly dependent on the domain (and the corresponding word sense). Russian words 'шифр', 'закон' or 'программа' would perhaps be optimal choices if we consider cryptography, legal and software development domains, respectively. However, the translation 'код' is also acceptable across all these domains and, as such, would be a safer choice when the target domain is unknown. Note that such a translation may not be the most frequent overall and, consequently, might not be proposed by a standard (i.e. domain-agnostic) phrase-based translation system.

In order to encode preference for domain-invariant translations, we can first *project* phrases onto a compact $(K-1)$ dimensional simplex of subdomains with vectors:

$$\tilde{\mathbf{e}} = \Big[P(z=1|\ \tilde{e}),\ \ldots,\ P(z=K|\ \tilde{e})\Big], \tag{45}$$

$$\tilde{\mathbf{f}} = \Big[P(z=1|\ \tilde{f}),\ \ldots,\ P(z=K|\ \tilde{f})\Big]. \tag{46}$$

See Fig. 7 for an illustration of the projection framework.

Of course, the subdomains are usually not specified in the heterogeneous training data. We can treat the subdomains as latent, and induce them automatically (Cuong et al, 2016). In the end, we can use a relevant measure to quantify how likely a phrase (or a phrase-pair) is to be "domain-invariant", for instance:

– *Domain-specificity of phrases*: A rule with source and target phrases having a *peaked* distribution over latent subdomains is likely domain-specific.

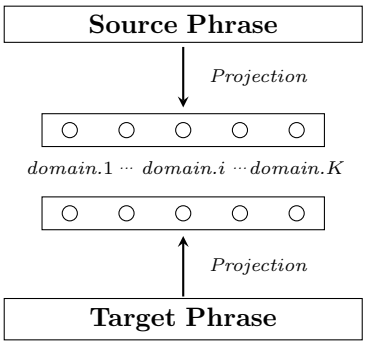

Fig. 7 The projection framework of phrases into K-dimensional vector space of probabilistic latent subdomains.

Technically speaking, entropy comes as a natural choice for quantifying domain specificity. Here, we opt for the Renyi entropy and define the domain specificity as follows:

$$D_\alpha(\tilde{\mathbf{e}}) = \frac{1}{1-\alpha} \log \left(\sum\nolimits_{i=1}^{K} P(z=i|\ \tilde{e})^\alpha \right) \tag{47}$$

$$D_\alpha(\tilde{\mathbf{f}}) = \frac{1}{1-\alpha} \log \left(\sum\nolimits_{i=1}^{K} P(z=i|\ \tilde{f})^\alpha \right) \tag{48}$$

Normally, the value of α is set as 2 which is the default choice (also known as the Collision entropy).

– *Source-target coherence across subdomains*: A translation rule with source and target phrases having two similar distributions over the latent subdomains is likely safer to use. We can use the Chebyshev distance for measuring the similarity between two distributions. The divergence of two vectors $\tilde{\mathbf{e}}$ and $\tilde{\mathbf{f}}$ is defined as follows

$$D(\tilde{\mathbf{e}},\ \tilde{\mathbf{f}}) = \max_{i=\{1,\ ...,\ K\}} \left| P(z=i|\ \tilde{e}) - P(z=i|\ \tilde{f}) \right| \tag{49}$$

The measures, once integrated into a phrase-based SMT system as feature functions, force the decoder to give higher preference to domain-invariant translations, which work well across domains, over risky domain-specific alternatives.

The translation improvement is quite robust: it is obtained without tuning specifically for the target domain or using other domain-related meta-information in the training corpus (Cuong et al, 2016).

A similar idea has been deployed in Zhang et al (2014), which exploits topic-insensitivity that is learned over documents for translation. There is a link between this line of work and extensive prior work on minimum Bayes risk (MBR) objectives (used either at test time (Kumar and Byrne, 2004) or during training (Goodman, 1998; Sima'an, 2003; Smith and Eisner, 2006; Pauls et al, 2009)). The goal of MBR minimization is to select translations

that are less "risky", but due to the uncertainty in model predictions, and some of this uncertainty may indeed be associated with domain-variability of translations. Still, a system trained with an MBR objective will tend to output most frequent translation rather than the most domain-invariant one, and this, as we argued in the introduction, might not be the right decision when applying it across domains. We believe that the two classes of methods are largely complementary.

9 Discussion

As discussed, SMT is just one among data-driven approaches to modeling translation. Other approaches can be deployed, e.g. example-based machine translation and neural-based machine translation. While it is pretty clear that example-based machine translation can benefit from what the domain adaptation literature for SMT offers, it would be less clear whether neural-based machine translation can learn from that or not. Recent studies suggest this is the case, where classic techniques in domain adaptation for SMT can be used to perform adaptation for neural-based translation models (e.g. see (Durrani et al, 2015; Joty et al, 2015)). More specifically, Durrani et al (2015) show that EM-based mixture modeling and data selection techniques also give a significant improvement in adaptation. Joty et al (2015) reveal that regularizing the loss function towards the in-domain neural network joint model also improves translation.

References

Axelrod A, He X, Gao J (2011) Domain adaptation via pseudo in-domain data selection. In: EMNLP

Aziz W, Dymetman M, Specia L (2014) Exact decoding for phrase-based statistical machine translation. In: EMNLP

Bertoldi N, Federico M (2009) Domain adaptation for statistical machine translation with monolingual resources. In: WMT

Besling S, Meier H (1995) Language model speaker adaptation. In: EUROSPEECH

Bhattacharyya A (1946) On a measure of divergence between two multinomial populations. Sankhyā: The Indian Journal of Statistics (1933-1960)

Birch A, Osborne M, Koehn P (2007) Ccg supertags in factored statistical machine translation. In: Proceedings of WMT

Bisazza A, Ruiz N, Federico M (2011) Fill-up versus interpolation methods for phrase-based smt adaptation. In: IWSLT

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. JMLR

Blunsom P, Osborne M (2008) Probabilistic inference for machine translation. In: EMNLP

Bod R, Scha R, Sima'an K (2003) Data-Oriented Parsing. Center for the Study of Language and Information - Lecture Notes, Center for the Study of Language and Inf

Brown PF, Cocke J, Pietra SAD, Pietra VJD, Jelinek F, Lafferty JD, Mercer RL, Roossin PS (1990) A statistical approach to machine translation. Comput Linguist

Brown PF, Pietra VJD, Pietra SAD, Mercer RL (1993) The mathematics of statistical machine translation: parameter estimation. Comput Linguist

Carpuat M, Goutte C, Foster G (2014) Linear mixture models for robust machine translation. In: WMT

Chang YW, Collins M (2011) Exact decoding of phrase-based translation models through lagrangian relaxation. In: Proceedings of EMNLP

Chang YW, Rush AM, DeNero J, Collins M (2014) A constrained viterbi relaxation for bidirectional word alignment. In: ACL

Chen B, Foster G, Kuhn R (2013a) Adaptation of reordering models for statistical machine translation. In: NAACL HLT

Chen B, Kuhn R, Foster G (2013b) Vector space model for adaptation in statistical machine translation. In: ACL

Chen B, Kuhn R, Foster G, Cherry C, Huang F (2016) Bilingual methods for adaptive training data selection for machine translation. AMTA

Cherry C, Foster G (2012) Batch tuning strategies for statistical machine translation. In: NAACL HLT

Chiang D (2005) A hierarchical phrase-based model for statistical machine translation. In: ACL

Chiang D (2007) Hierarchical phrase-based translation. Comput Linguist

Chiang D, Marton Y, Resnik P (2008) Online large-margin training of syntactic and structural translation features. In: EMNLP

Chiang D, Knight K, Wang W (2009) 11,001 new features for statistical machine translation. In: NAACL HLT

Chiang D, DeNeefe S, Pust M (2011) Two easy improvements to lexical weighting. In: ACL HLT (Short Papers)

Clark J, Dyer C, Lavie A (2014) Locally non-linear learning for statistical machine translation via discretization and structured regularization. TACL

Clarkson P, Robinson A (1997) Language model adaptation using mixtures and an exponentially decaying cache. In: ICASSP

Cui L, Chen X, Zhang D, Liu S, Li M, Zhou M (2013) Multi-domain adaptation for SMT using multi-task learning. In: EMNLP

Cuong H, Sima'an K (2014a) Latent domain phrase-based models for adaptation. In: EMNLP

Cuong H, Sima'an K (2014b) Latent domain translation models in mix-of-domains haystack. In: COLING

Cuong H, Sima'an K (2015) Latent domain word alignment for heterogeneous corpora. In: NAACL-HLT

Cuong H, Sima'an K, Titov I (2016) Adapting to all domains at once: Rewarding domain invariance in smt. In: TACL

Daumé H III, Jagarlamudi J (2011) Domain adaptation for machine translation by mining unseen words. In: ACL-HLT (short papers)
Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B 39(1):1–38
Devlin J, Zbib R, Huang Z, Lamar T, Schwartz R, Makhoul J (2014) Fast and robust neural network joint models for statistical machine translation. In: ACL
Dong M, Cheng Y, Liu Y, Xu J, Sun M, Izuha T, Hao J (2014) Query lattice for translation retrieval. In: COLING
Duh K, Sudoh K, Tsukada H (2010) Analysis of translation model adaptation in statistical machine translation. In: IWSLT
Duh K, Neubig G, Sudoh K, Tsukada H (2013) Adaptation data selection using neural language models: Experiments in machine translation. In: ACL (Short Papers)
Durrani N, Sajjad H, Joty S, Abdelali A, Vogel S (2015) Using joint models for domain adaptation in statistical machine translation. In: MTSummit
Eck M, Vogel S, Waibel A (2005) Low cost portability for statistical machine translation based on n-gram coverage. In: MTSummit
Eidelman V, Boyd-Graber J, Resnik P (2012) Topic models for dynamic translation model adaptation. In: ACL (Short Papers)
Federico M, Cettolo M, Bentivogli L, Paul M, Stüker S (2012) Overview of the IWSLT 2012 evaluation campaign. In: IWSLT
Foster G, Kuhn R (2007) Mixture-model adaptation for smt. In: WMT
Foster G, Goutte C, Kuhn R (2010) Discriminative instance weighting for domain adaptation in statistical machine translation. In: EMNLP
Foster G, Chen B, Kuhn R (2013) Simulating discriminative training for linear mixture adaptation in statistical machine translation. In: MTSummit
Galley M, Manning CD (2008) A simple and effective hierarchical phrase reordering model. In: EMNLP
Gao Q, Lewis W, Quirk C, Hwang MY (2011) Incremental training and intentional over-fitting of word alignment. In: MT Summit
Gong Z, Zhang M, Zhou G (2011) Cache-based document-level statistical machine translation. In: EMNLP
Goodman JT (1998) Parsing inside-out. PhD thesis
Green S, Wang S, Cer DM, Manning CD (2013) Fast and adaptive online training of feature-rich translation models. In: ACL
Green S, Cer D, Manning C (2014) An empirical comparison of features and tuning for phrase-based machine translation. In: WMT
Gruber A, Weiss Y, Rosen-zvi M (2007) Hidden topic markov models. In: AISTATS, Journal of Machine Learning Research - Proceedings Track
Haddow B (2013) Applying pairwise ranked optimisation to improve the interpolation of translation models. In: NAACL HLT (Short Papers)
Haghighi A, Liang P, Berg-Kirkpatrick T, Klein D (2008) Learning bilingual lexicons from monolingual corpora. In: ACL

Hasler E, Haddow B, Koehn P (2012) Sparse lexicalised features and topic adaptation for SMT. In: IWSLT
Hasler E, Blunsom P, Koehn P, Haddow B (2014) Dynamic topic adaptation for phrase-based mt. In: EACL
Hewavitharana S, Mehay D, Ananthakrishnan S, Natarajan P (2013) Incremental topic-based translation model adaptation for conversational spoken language translation. In: ACL (Short Papers)
Hieber F, Riezler S (2015) Bag-of-words forced decoding for cross-lingual information retrieval. In: HLT-NAACL
Hofmann T (1999) Probabilistic latent semantic analysis. In: UAI
Hopkins M, May J (2011) Tuning as ranking. In: EMNLP
Hu Y, Zhai K, Eidelman V, Boyd-Graber J (2014) Polylingual tree-based topic models for translation domain adaptation. In: ACL
Hua W, Haifeng W, Zhanyi L (2005) Alignment model adaptation for domain-specific word alignment. In: ACL
Irvine A, Morgan J, Carpuat M, III DH, Munteanu D (2013a) Measuring machine translation errors in new domains. In: TACL
Irvine A, Quirk C, III HD (2013b) Monolingual marginal matching for translation model adaptation. In: EMNLP
Jeblee S, Feely W, Bouamor H, Lavie A, Habash N, Oflazer K (2014) Domain and dialect adaptation for machine translation into egyptian arabic. In: EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)
Joty S, Sajjad H, Durrani N, Al-Mannai K, Abdelali A, Vogel S (2015) How to avoid unwanted pregnancies: Domain adaptation using neural network models. In: EMNLP
Kettunen K (2009) Choosing the best mt programs for clir purposes — can mt metrics be helpful? In: ECIR
Kirchhoff K, Bilmes J (2014) Submodularity for data selection in machine translation. In: EMNLP
Koehn P (2004) Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In: Machine Translation: From Real Users to Research
Koehn P (2010) Statistical Machine Translation. Cambridge University Press
Koehn P, Knight K (2002) Learning a translation lexicon from monolingual corpora. In: ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9
Koehn P, Schroeder J (2007) Experiments in domain adaptation for statistical machine translation. In: WMT
Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. In: NAACL HLT
Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: Open source toolkit for statistical machine translation. In: ACL on Interactive Poster and Demonstration Sessions
Kuhn R, De Mori R (1990) A cache-based natural language model for speech recognition. PAMI

Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Statist

Kumar S, Byrne WJ (2004) Minimum bayes-risk decoding for statistical machine translation. In: HLT-NAACL

Lambert P, Schwenk H, Servan C, Abdul-Rauf S (2011) Investigations on translation model adaptation using monolingual data. In: WMT

Lewis W, Eetemadi S (2013) Dramatically Reducing Training Data Size Through Vocabulary Saturation. In: WMT

Liu C, Liu Y, Sun M, Luan H, Yu H (2015) Generalized agreement for bidirectional word alignment. In: EMNLP

Liu L, Watanabe T, Sumita E, Zhao T (2013) Additive neural networks for statistical machine translation. In: ACL

Lopez A (2008) Statistical machine translation. ACM Comput Surv

Louis A, Webber BL (2014) Structured and unstructured cache models for SMT domain adaptation. In: EACL

Macherey W, Och FJ, Thayer I, Uszkoreit J (2008) Lattice-based minimum error rate training for statistical machine translation. In: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '08, pp 725–734, URL http://dl.acm.org/citation.cfm?id=1613715.1613807

Mansour S, Ney H (2014) Unsupervised adaptation for statistical machine translation. In: Proceedings of WMT

Mansour S, Wuebker J, Ney H (2011) Combining Translation and Language Model Scoring for Domain-Specific Data Filtering. In: IWSLT

Marton Y, Resnik P (2008) Soft syntactic constraints for hierarchical phrased-based translation. In: ACL

Matsoukas S, Rosti AVI, Zhang B (2009) Discriminative corpus weight estimation for machine translation. In: EMNLP

Mimno D, Wallach H, Naradowsky J, Smith DA, McCallum A (2009) Polylingual topic models. In: EMNLP

Moore RC, Lewis W (2010) Intelligent selection of language model training data. In: ACL (Short Papers)

Nakov P (2008) Improving english-spanish statistical machine translation: Experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In: WMT

Neubig G, Watanabe T (2016) Optimization for statistical machine translation: A survey. Comput Linguist

Nikoulina V, Kovachev B, Lagos N, Monz C (2012) Adaptation of statistical machine translation model for cross-lingual information retrieval in a service context. In: EACL

Och F (2003) Minimum error rate training in statistical machine translation. In: ACL

Och F, Ney H (2004) The alignment template approach to statistical machine translation. Comput Linguist

Och FJ, Ney H (2002) Discriminative training and maximum entropy models for statistical machine translation. In: ACL

Pauls A, DeNero J, Klein D (2009) Consensus training for consensus decoding in machine translation. In: EMNLP

Pecina P, Toral A, van Genabith J (2012) Simple and effective parameter tuning for domain adaptation of statistical machine translation. In: COLING

Quirk C, Menezes A (2006) Dependency treelet translation: The convergence of statistical and example-based machine-translation? Machine Translation 20(1):43–65, DOI 10.1007/s10590-006-9008-4, URL http://dx.doi.org/10.1007/s10590-006-9008-4

Quirk C, Menezes A, Cherry C (2005) Dependency treelet translation: Syntactically informed phrasal smt. In: ACL

Razmara M, Foster G, Sankaran B, Sarkar A (2012) Mixing multiple translation models in statistical machine translation. In: ACL

Schwenk H (2008) Investigations on large-scale lightly-supervised training for statistical machine translation. In: IWSLT

Schwenk H, Senellart J (2009) Translation model adaptation for an arabic/french news translation system by lightly-supervised training. In: MT Summit

Sennrich R (2012) Perplexity minimization for translation model domain adaptation in statistical machine translation. In: EACL

Shah K, Barrault L, Schwenk H (2010) Translation model adaptation by resampling. In: WMT

Shah K, Barrault L, Schwenk H (2012) A general framework to weight heterogeneous parallel data for model adaptation in statistical machine translation. In: Proceedings of AMTA

Shen S, Liu Y, Sun M, Luan H (2015) Consistency-aware search for word alignment. In: EMNLP

Sima'an K (2003) On maximizing metrics for syntactic disambiguation. In: IWPT

Simianer P, Riezler S, Dyer C (2012) Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In: ACL

Simion A, Collins M, Stein C (2013) A convex alternative to ibm model 2. EMNLP

Smith DA, Eisner J (2006) Minimum risk annealing for training log-linear models. In: COLING-ACL

Snover M, Dorr B, Schwartz R (2008) Language and translation model adaptation using comparable corpora. In: EMNLP

Su J, Wu H, Wang H, Chen Y, Shi X, Dong H, Liu Q (2012) Translation model adaptation for statistical machine translation with monolingual topic information. In: ACL

Tamura A, Watanabe T, Sumita E (2012) Bilingual lexicon extraction from comparable corpora using label propagation. In: EMNLP-CoNLL

Tamura A, Watanabe T, Sumita E (2014) Recurrent neural networks for word alignment model. In: ACL

Tang J, Meng Z, Nguyen X, Mei Q, Zhang M (2014) Understanding the limiting factors of topic modeling via posterior contraction analysis. In: ICML, pp 190–198

Tiedemann J (2010) Context adaptation in statistical machine translation using models with exponentially decaying cache. In: 2010 Workshop on Domain Adaptation for Natural Language Processing

Tillmann C (2004) A unigram orientation model for statistical machine translation. In: HLT-NAACL (Short Papers)

Tsuruoka Y, Tsujii J, Ananiadou S (2009) Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In: ACL

Vogel S, Ney H, Tillmann C (1996) Hmm-based word alignment in statistical translation. In: COLING, pp 836–841

Waite A, Byrne W (2015) The geometry of statistical machine translation. In: NAACL

Wang X, Utiyama M, Finch A, Watanabe T, Sumita E (2015) Leave-one-out word alignment without garbage collector effects. In: EMNLP

Wäschle K, Riezler S (2012) Structural and topical dimensions in multi-task patent translation. In: EACL

Watanabe T, Suzuki J, Tsukada H, Isozaki H (2007) Online large-margin training for statistical machine translation. In: EMNLP-CoNLL

Wees Mvd, Bisazza A, Weerkamp W, Monz C (2015) What's in a domain? analyzing genre and topic differences in smt. In: Proceedings of ACL-IJCNLP (short paper)

Wu H, Wang H, Zong C (2008) Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In: COLING

Yamada K, Knight K (2001) A syntax-based statistical translation model. In: 39th Annual Meeting on Association for Computational Linguistics, ACL

Yu H, Huang L, Mi H, Zhao K (2013) Max-violation perceptron and forced decoding for scalable mt training. In: EMNLP

Zhang B, Su J, Xiong D, Duan H, Yao J (2015) Discriminative reordering model adaptation via structural learning. In: IJCAI

Zhang H, Chiang D (2014) Kneser-ney smoothing on expected counts. In: Proceedings of ACL

Zhang M, Xiao X, Xiong D, Liu Q (2014) Topic-based dissimilarity and sensitivity models for translation rule selection. JAIR